

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s): KAGEYAMA, et al.
Serial No.: Not yet assigned
Filed: January 21, 2004
Title: AUTOMATIC TEST PROGRAM GENERATION METHOD
Group: Not yet assigned

LETTER CLAIMING RIGHT OF PRIORITY

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

January 21, 2004

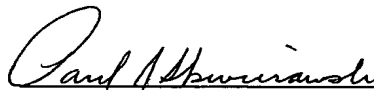
Sir:

Under the provisions of 35 USC 119 and 37 CFR 1.55, the applicant(s) hereby claim(s) the right of priority based on Japanese Patent Application No.(s) 2003-011970, filed January 21, 2003.

A certified copy of said Japanese Application is attached.

Respectfully submitted,

ANTONELLI, TERRY, STOUT & KRAUS, LLP



Paul J. Skwierawski
Registration No. 32,173

PJS/alb
Attachment
(703) 312-6600

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: 2003年 1月21日

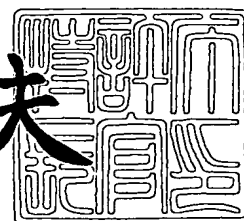
出願番号
Application Number: 特願2003-011970
[ST. 10/C]: [JP2003-011970]

出願人
Applicant(s): 株式会社日立製作所
株式会社日立インフォメーションテクノロジー

2003年12月25日

特許庁長官
Commissioner,
Japan Patent Office

今井康夫



出証番号 出証特2003-3107547



【書類名】 特許願

【整理番号】 NT02P0762

【提出日】 平成15年 1月21日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 9/06

【発明者】

【住所又は居所】 神奈川県秦野市堀山下 1 番地 株式会社日立製作所 エ
ンタープライズサーバ事業部内

【氏名】 影山 直洋

【発明者】

【住所又は居所】 神奈川県秦野市堀山下 1 番地 株式会社日立製作所 エ
ンタープライズサーバ事業部内

【氏名】 廣瀬 善太郎

【発明者】

【住所又は居所】 東京都小平市上水本町五丁目 2 0 番 1 号 株式会社日立
製作所 半導体グループ内

【氏名】 河内 満幸

【発明者】

【住所又は居所】 神奈川県足柄上郡中井町境 4 5 6 番地 株式会社日立イ
ンフォメーションテクノロジー内

【氏名】 福真 武

【特許出願人】

【識別番号】 000005108

【氏名又は名称】 株式会社日立製作所

【特許出願人】

【識別番号】 000153454

【氏名又は名称】 株式会社日立インフォメーションテクノロジー

【代理人】

【識別番号】 100068504
【弁理士】
【氏名又は名称】 小川 勝男
【電話番号】 03-3661-0071

【選任した代理人】

【識別番号】 100086656
【弁理士】
【氏名又は名称】 田中 恭助
【電話番号】 03-3661-0071

【手数料の表示】

【予納台帳番号】 081423
【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1
【物件名】 図面 1
【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 コンパイラテストプログラムの自動生成方法

【特許請求の範囲】

【請求項 1】

コンパイラのテストプログラムの要素となる小手続き記述の集合から乱数を用いて該小手続き記述を複数選択し、選択された小手続き記述を組み合わせて、手続き記述を生成することを特徴とするテストプログラムの自動生成方法。

【請求項 2】

乱数を用いて小手続き記述を選択する場合、前記小手続き記述の選択のされ易さを示す数値を該小手続き記述に割り当て、選択に該数値を反映することを特徴とする請求項 1 記載のテストプログラムの自動生成方法。

【請求項 3】

選択された複数の小手続き記述の間で、該小手続き記述に含まれる変数を共用化し、共用化された変数を持つ小手続き記述が組み合わせられることを特徴とする請求項 1 記載のテストプログラムの自動生成方法。

【請求項 4】

選択対象となる小手続き記述自身が、その小手続き記述を計算機で実行した時の動作結果を計算機の表示装置に表示する手続きおよび／又は記憶装置に格納する手続きを有することを特徴とする請求項 1 記載のテストプログラムの自動生成方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、テストプログラムの作成に係り、さらに詳しくは、コンパイラ開発時のテストに用いるテストプログラムの作成に適したテストプログラムの作成に関する。

【0002】

【従来の技術】

従来、高級言語で記述されたプログラムを計算機上で実行可能な形式に翻訳す

るコンパイラの開発において、その機能を検証するのに用いるテストプログラムは、コンパイラ設計者がその処理内容を考慮して人手作成したり、プログラム言語仕様に詳しい者が、その仕様に基づき人手により作成したりしていた。

【0 0 0 3】

また、その一方で、テストプログラムの自動作成を目指し、計算機によるテストプログラムの自動生成も考えられている。その一つは、用意された複数の構文生成規則から乱数を用いて規則を選択し、選択された規則を基にテストプログラムを生成するものである（例えば、特許文献 1）。また、既存のテストプログラムの文法網羅性を判定し、既存のテストプログラムから外れた文法を縮小仕様文法として出力し、その縮小文法仕様を基に追加テストプログラムを自動的に生成するものがある（例えば、特許文献 2）。更に、バックス記法で記述された構文規則を基に文パターンを生成し、生成された文パターンを人手によって編集し、テストプログラムとするものがある（例えば、特許文献 3）。

【0 0 0 4】

これらのテストプログラム自動生成方法は、いずれもテストプログラム内の単一の文を自動生成するのがその機能である。

【0 0 0 5】

【特許文献 1】

特開平 1 - 2 2 0 0 4 4 号公報

【特許文献 2】

特開平 3 - 7 5 8 4 0 号公報

【特許文献 3】

特開平 6 - 2 1 4 7 7 5 号公報

【0 0 0 6】

【発明が解決しようとする課題】

コンパイラの機能不良を摘出するには、テストプログラムを多数作成しテストを実施する必要がある。しかも近年の計算機アーキテクチャの複雑化に伴い、コンパイラ機能も複雑化し、コンパイラの全機能を十分にテストするには、上記従来のように人手によりテストプログラムを作成していたのでは、その工数は膨大

になる。従って、コンパイラの出荷前に十分な機能検証が出来ない惧れがある。

【0007】

これに対処するため、上述のような計算機によるテストプログラムの自動生成方法が提案されてきた。しかしながら、上述の方法はテストプログラムの単一の文を対象としている。摘出困難で重大な機能上の不具合は一般にはプログラム構造が複雑な場合であり、単一文内に閉じて発生するものは少なく、複数の文にまたがって条件が重なったときに発生する。この種の機能上の不具合は従来の主に単一文を生成対象としてきた計算機による自動生成方法では、摘出するのは困難である。

【0008】

【課題を解決するための手段】

本発明のテストプログラムの自動生成方法では、複数のプログラムの部分的な記述(小手続き記述、プログラムセルと呼ぶテストプログラムの要素)を、乱数を用いて選択し、組み合わせて、テストプログラムを生成する。

【0009】

【発明の実施の形態】

以下、本発明の実施例を図面を用いて説明する。

図1は、本発明のテストプログラム自動生成方法を実施してコンパイラのテストプログラムを自動生成するテストプログラム発生器100の構成例を示した図である。本実施例で示すテストプログラム発生器100は、例えばパーソナルコンピュータ上で動作するソフトウェアにより実現される。

【0010】

図1において、プログラムセル(単にセルとも言う)情報110とは、複数のプログラムの部分的な記述に関する情報の集まりであり、コンピュータの記憶装置に格納される。プログラムセルとは、図3、図4、図5図11を用いて詳細に後述するが、テストプログラムを構成する構成要素であり、各セルはセル重み情報111にてセル重みが規定されている。セル重みは、プログラムセルの集まりから特定のプログラムセルを乱数を用いて選択する時に、セル重みが重いほど当該セルが選択され易くなるよう取り扱われる。乱数発生器120により発生された乱数を

基にプログラムセル組合せ器130にて、プログラムセルを選択し、該プログラムセルを組み合わせてテストプログラム140を生成する。

【0011】

重み付けは、例えば、以前にテストプログラムのバグを発見出来たセルや、バグを発見出来そうと予測されるセルに対して与えられる。重み付けはテスト毎に変更可能である。

【0012】

図2は、図1で説明したテストプログラム発生器100を組み込んだコンパイラ検査システムの構成例を示した図である。本システムは、例えばパーソナルコンピュータ上で動作するソフトウェアにより実現される。該テストプログラム発生器100の出力であるテストプログラム140をテスト実行器220に入力する。テスト実行器220内では、検証対象コンパイラに対してテストプログラムを入力し、コンパイルを実施する230。コンパイルをして得られる実行形式ファイルを実行し240、その結果を実行結果260として記憶装置に格納する。

【0013】

他方、同一のテストプログラム140を期待値発生器210に入力する。期待値発生器210は、例えば使用実績のあるコンパイラを用いてテストプログラムをコンパイルし、コンパイルして得られる実行形式ファイルを実行することで実現できる。テストプログラム140を期待値発生器210に入力して得られた期待値250と該実行結果260とを結果比較器270に入力し、比較した結果を検査結果280として計算機の記憶装置に格納する。

【0014】

セル単体のテストは図7に後述のテストプログラムの作成に先立ってセル自体の正しさをチェックするためのものである。上記のテストプログラムの検査結果は複数のセルを組み合わせて1つのテストプログラムとして実行させた結果であるが、どのセルで異常があったかは分かるようになっている。

【0015】

図3から図6、および図11は本発明で提案するプログラムセルの一例を、PAD図を用いて示したものである。以下ではプログラムセルを基本セル、制御セル

、関数セル、宣言・定義セルの 4 種類に分類し説明する。

【0 0 1 6】

図 3 は、基本セルの構成例を示した PAD 図である。基本セルは、「変数他の定義と初期値設定」320 と「実行文」330 と「実行結果の表示、保存」340 が実行順序に従って記述されているプログラムセルである。さらに基本セルを単体でコンパイル・実行する場合に必要な「単体動作時に必要な記述 1」310 と「単体動作時に必要な記述 2」350 を含む。コンパイル時のオプション指定により「単体動作時に必要な記述 1」310 と「単体動作時に必要な記述 2」350 を制御し、基本セル単体でのコンパイルと動作確認を可能にする。

「単体動作時に必要な記述 1」310 は、例えば、

```
#ifdef _CELL_COMPILE_CHECK_
    void main() {
        #endif
```

と記述できる。また「単体動作時に必要な記述 2」350 は、例えば

```
#ifdef _CELL_COMPILE_CHECK_
    {
        #endif
```

と記述できる。トークン `_CELL_COMPILE_CHECK_` が定義されている時のみセル単体でのコンパイル・実行が可能である。プログラムセルを組み合わせたテストプログラムをコンパイル・実行する時は、トークン `_CELL_COMPILE_CHECK_` を定義しない。

【0 0 1 7】

「変数他の定義と初期値設定」320 では該基本セルを計算機上で翻訳・実行する場合に必要な変数他の定義と初期値設定のための記述である。変数の定義のほか定数やマクロなどの定義も含まれる。

【0 0 1 8】

「実行結果の表示、保存」340 記述は、基本セルの実行結果を計算機に接続する表示装置に表示し、また記憶装置に格納するための記述である。これによりプログラムセルが組込まれたテストプログラムを実行した場合、テスト実行者がテ

スト実行時にテストプログラムの動作状況を逐一把握できる。

【0019】

図4と図5は、制御セルの構成例を示したPAD図である。本実施例では、制御セルを図4に示す選択セルと図5に示す反復セルに分類した。いずれも「単体動作時に必要な記述1」310と「単体動作時に必要な記述2」350が、記述として組込まれているため、コンパイル時にオプションを指定することにより「単体動作時に必要な記述1」310と「単体動作時に必要な記述2」350を制御し、制御セル単体でのコンパイルと動作を確認することができる。

【0020】

図4は選択セルの構成例を示したPAD図である。「変数他の定義と初期値設定」410は、基本セルの場合と同様に、該選択セルを計算機上で翻訳・実行する場合に必要な変数他の定義と初期値設定のための記述である。「選択」420は、条件の記述と該条件に応じて「実行文」430を起動する記述である。選択セルには、「他のプログラムセル」440を入れ子で組み込むことが可能である。他のプログラムセルが組み込まれていなくともこの選択セルは実行出来る。「実行結果の表示、保存」450は、基本セルと同様に、選択セルの実行結果を計算機に接続する表示装置に表示し、また記憶装置に格納するための記述である。

【0021】

図5は反復セルの構造を示したPAD図である。「変数他の定義と初期値設定」510は、基本セルや制御セルと同様に、該反復セルを計算機上で翻訳・実行する場合に必要な変数他の定義と初期値設定のための記述である。「反復」520は、「実行文」530を反復して起動するための記述である。反復セルには、選択セルと同様「他のプログラムセル」540を入れ子で組込むことが可能である。「実行結果の表示、保存」550は、基本セルや選択セルと同様に、反復セルの実行結果を計算機に接続する表示装置に表示し、また記憶装置に格納するための記述である。

【0022】

図6は関数セルの構造を示したPAD図である。関数セルも基本セルや制御セルと同様に単独で動作させることが可能である。「単体動作時に必要な記述1」310と「単体動作時に必要な記述2」350が組込まれているため、コンパイル時のオ

プシオン指定により「単体動作時に必要な記述 1」310と「単体動作時に必要な記述2」350を制御し、関数セル単体でのコンパイルと動作を確認することができる。基本セルや制御セルと同様に「変数他の定義と初期値設定」610は、該関数セルを計算機上で翻訳・実行する場合に必要な変数他の定義と初期値設定のための記述である。「関数」620は、よそで定義された関数を呼び出すための記述である。「実行結果の表示、保存」630は、基本セルや制御セルと同様に、該関数セルの実行結果を計算機に接続する表示装置に表示し、また記憶装置に格納するための記述である。

【 0 0 2 3 】

図 1 1 は宣言・定義セルの構造の一例を示した P A D 図である。宣言・定義セルは、他のプログラムセル間で共通に利用する構造体等の宣言や関数セルの関数の定義、`#include`文を記述する。宣言・定義セルは、プログラムセルを組み合わせる時に必ず先頭に一つ配置される。

【 0 0 2 4 】

図 7 は、図 1 に示すプログラムセル組合せ器 130 で実施する処理の処理手順を示した図である。計算機上の記憶装置に格納された図 3 から図 6 および図 1 1 に示す基本セル、制御セル、関数セル、宣言・定義セルの各プログラムセル情報 110 とセル重み情報 111 を入力として、乱数発生器 120 で生成された乱数を用いてプログラムセルを選択し、組み合わせてテストプログラム 140 を生成する。詳細は後述する。

【 0 0 2 5 】

図 8、図 9、図 1 0 は、図 7 に示す処理手順を計算機のソフトウェアを用いて実現する場合に、計算機の主記憶上に実現されるテーブルの一例を示す。図 8 はセル管理テーブル 800 の一例を示した図である。項番 810 はセルのエントリ番号が格納される。セル名称 820 にはセルの名称が格納される。セルタイプ 830 には、基本セル／制御セル／関数セル／宣言・定義セルのセルの種類が格納される。

【 0 0 2 6 】

セル重み 840 は、セル管理テーブルに登録されたセルの中から、乱数を用いてセルを選択する時に選択され易さを反映させるための値である。例えばセル重み

2のプログラムセルは、セル重み1のプログラムセルに比較して2倍選択され易くする等が考えられる。セル記述テーブル情報850は、セル記述テーブル900に関する情報を格納する。セル記述テーブル情報850は、ポインタ851と記述長852とから構成される。ポインタ851は、該当セルのセル記述テーブル900内でのセル記述の先頭文字の番地を示し、記述長852は該セル記述の長さを文字数で示す。

【 0 0 2 7 】

図 9 は、セル記述テーブル900の一例を示した図である。セル記述テーブル900は、番地910とセル記述文字920とからなる。番地910はセル記述テーブル900に格納される文字のアドレスを示し、文字920には、セル記述の各文字が格納される。例えば、図 8 の項番 1 のセルの実際の記述そのものは、セル記述テーブル情報 8 5 0 にあるようにポインタ 8 5 1 が 0 を指し、記述長 8 5 2 が 1 5 6 である。すると、図 8 の項番 1 の記述は図 9 の 0 番地から 1 5 5 番地までにあるということが分かる。ここで、文字にはセル記述に使われている英数字、空白、改行記号などセル記述に必要な情報が全て含まれる。従って図 8 の項番 1 のセルを実行するときは図 9 の 0 番地から 1 5 5 番地まで読み出される。

【 0 0 2 8 】

図 1 0 は、選択セルテーブル1000の構成例を示した図である。選択セルテーブル1000は、実行順番1010と選択セルポインタ1020とからなる。実行順番1010は、図 7 に示す処理手順に従って、乱数を用いて選択されたプログラムセルの選択テーブル内でのエントリ順（テストプログラム中のセルの並び順）を示す。選択セルポインタ1020は、図 7 に示す処理手順に従って、乱数を用いて選択されたプログラムセルのセル管理テーブル800の項番が格納される。

【 0 0 2 9 】

以下、図7に示す処理手順に従って、図 8 から図 1 0 に示すテーブルを用いてプログラムセル組合せ器で実現する処理内容について説明する。

【 0 0 3 0 】

図 7 に示す処理手順において、「内部テーブルへの展開」710で、図 1 のプログラムセル情報110とセル重み情報111は、図 8 に示すセル管理テーブル800と図 9 に示すセル記述テーブル900に展開される。「乱数を用いたプログラムセル列

の生成」720では、図8に示すセル管理テーブル800に登録されたプログラムセルを対象に、乱数発生器120で発生させた乱数を用いてプログラムセルを選択する。この時セル管理テーブル800のセル重み840に記載された値を基に、該当するプログラムセルの選択され易さを調整する。例えばセル重み3のプログラムセルの選択され易さをセル重み1のプログラムセルの3倍となる様に調整することができる。選択されたプログラムセルは選択セルテーブル1000に登録される。即ち、実行順番1010にセルが選択された順を、選択セルポインタ1020に選択されたプログラムセルのセル管理テーブル800内の項番を格納する。

【0031】

選択セルテーブル1000に登録された順序に従って、選択セルテーブル1000に登録されたプログラムセルの最後まで、プログラムセルに対して以下の処理を実施する730。着目したプログラムセルの一つ手前のセルが、基本／関数セルか制御セルかの判定740を行ない、基本セルと関数セルの場合には、着目したプログラムセルを接続して配置750する。制御セルの場合は、入れ子構造となるよう着目したプログラムセルを組込む760。

【0032】

以上の手順に従い組立てられたプログラムセルで構成されるテストプログラムにおいて、プログラムセル間での変数の共用化を行なう770。変数の共用化とはセル間で変数の名称を同じにして、前のセルの演算結果が後のセルの演算に反映できるようにすることを示す。プログラムセルは、それぞれ独自の変数を定義しているが、ここでは変数を共用化しプログラムセル間でデータのやり取りが行なえるようにする。

【0033】

以上の処理手順に従いテストプログラムを自動生成することが可能である。

【0034】

以上のように、セルを種類毎に分け、これらから実行するセルとその実行順を決めるようにしたため、コンパイラのテストに必要な多様なテストプログラムが出来る。

【0035】

本実施例によれば、複数のプログラムセルから、プログラムセルに付属するセル重みを基に、乱数を用いてプログラムセルを順序付けて選択し、組み合わせることにより、テストプログラムセルを自動生成することができる。これによりコンパイラをテストするに十分なテストプログラムを得ることができる。さらに、セル重みを調整することで、選択されるプログラムセルの傾向を変えることができ、例えば不良の発生傾向に応じて生成されるテストプログラムを調整することができる。従って、テストプログラム作成自動化によりテスト作成工数削減とコンパイラの網羅的なテストを可能にする。

【0036】

付記

1. コンパイラによって翻訳され前記コンパイラのテストプログラムの要素となる小手続き記述の集合から乱数を用いて該小手続き記述を複数選択し、選択された小手続き記述を組み合わせ、手続き記述を生成するテストプログラムの自動生成方法であって、

前記小手続き記述自身が、該小手続き記述を翻訳する場合、単独で翻訳する場合と複数の小手続き記述を組み合わせ生成された新たな手続き記述を翻訳する場合を切り替えるための記述を有することを特徴とするテストプログラムの自動生成方法。

2. コンパイラによって翻訳され前記コンパイラのテストプログラムの要素となる小手続き記述の集合から乱数を用いて該小手続き記述を複数選択し、選択された小手続き記述を組み合わせ、手続き記述を生成するテストプログラムの自動生成方法であって、

前記小手続き記述は、手続きを実行順序に従って並べた小手続き記述と、手続きの繰返しを記述した小手続き記述と、選択を記述した小手続き記述とを有することを特徴とするテストプログラムの自動生成方法。

【0037】

【発明の効果】

本発明によれば、多様なテストプログラムを自動的に作成することが出来る。

【図面の簡単な説明】

【図 1】

テストプログラム発生器の構成例を示す図である。

【図 2】

コンパイラ検査システムの構成例を示す図である。

【図 3】

基本セルの構成例を示す図である。

【図 4】

選択セルの構成例を示す図である。

【図 5】

反復セルの構成例を示す図である。

【図 6】

関数セルの構成例を示す図である。

【図 7】

プログラムセル組合せ器の処理手順の一例を示す図である。

【図 8】

セル管理テーブルの構成例を示す図である。

【図 9】

セル記述テーブルの構成例を示す図である。

【図 1 0】

選択セルテーブルの構成例を示す図である。

【図 1 1】

宣言・定義セルの構成を示す図である。

【符号の説明】

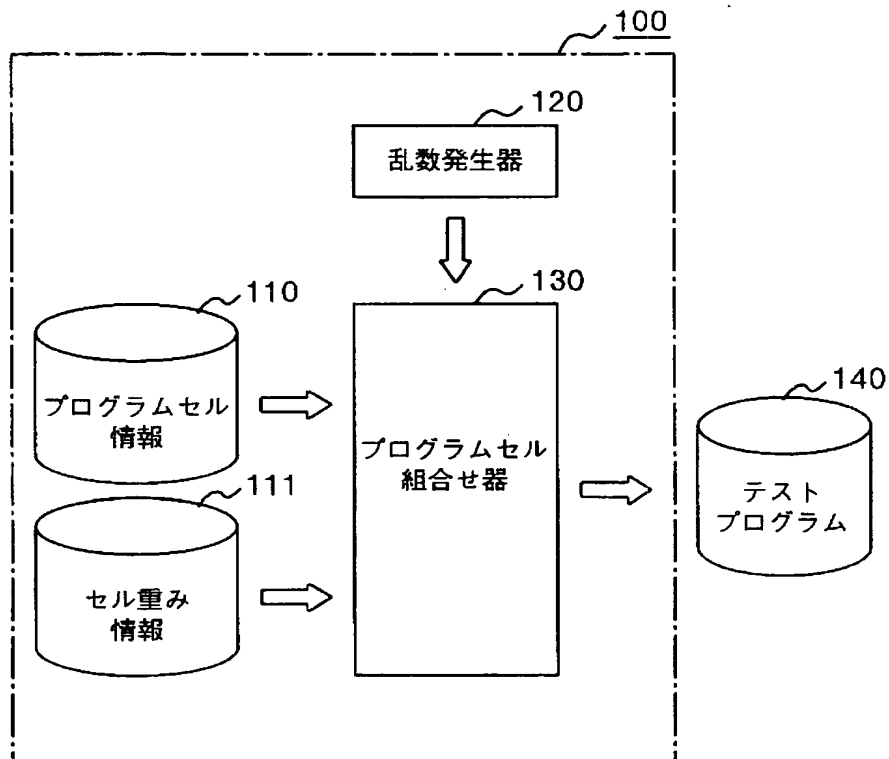
110…プログラムセル情報、111…セル重み情報、120…乱数発生器

130…プログラムセル組合せ器、140…テストプログラム

【書類名】 図面

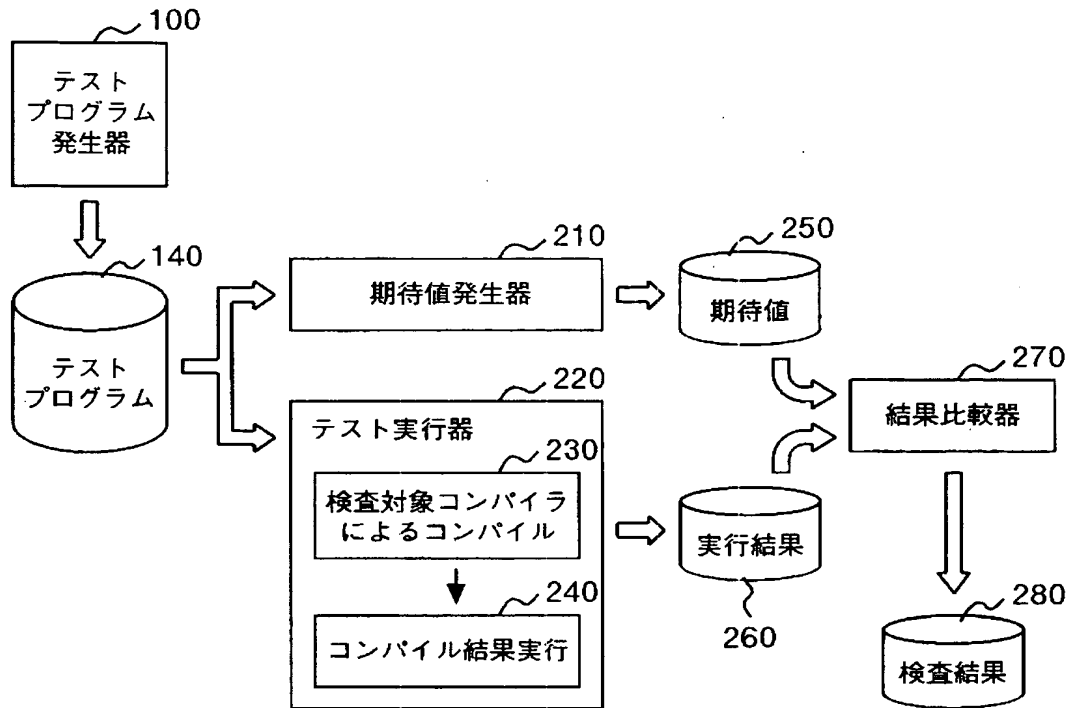
【図 1】

図 1



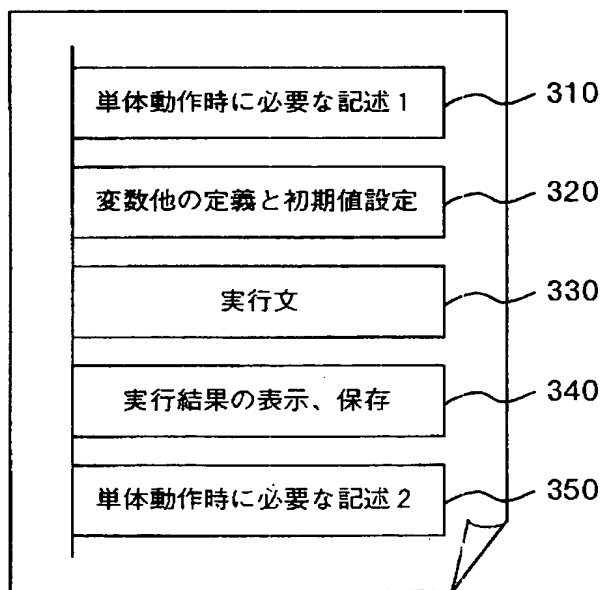
【図 2】

図 2



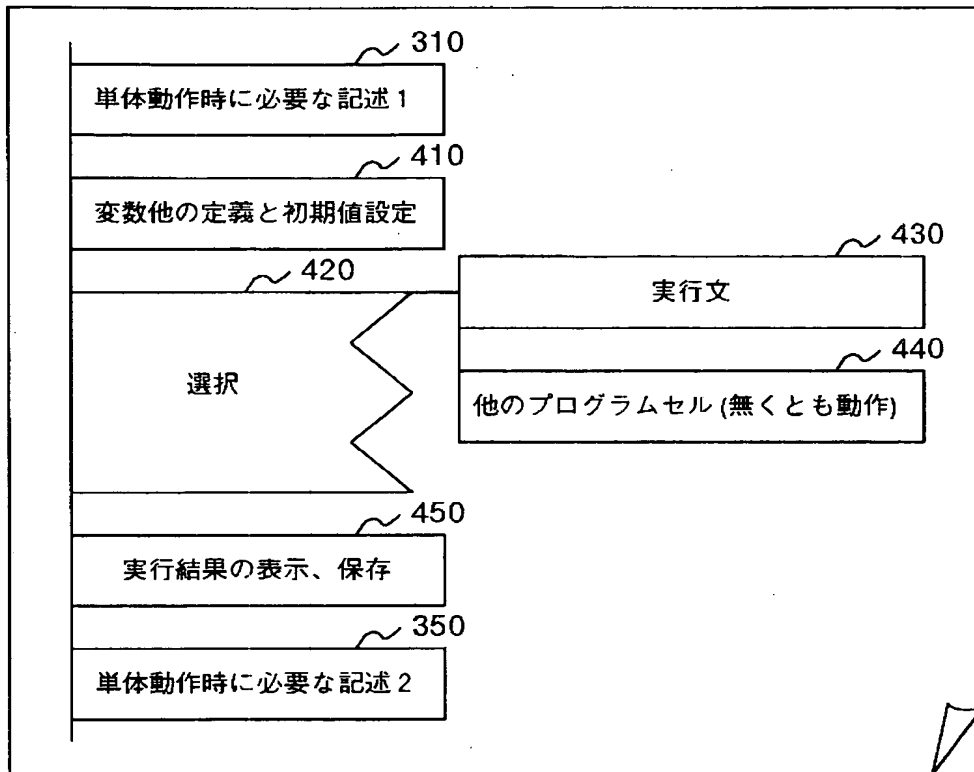
【図 3】

図 3



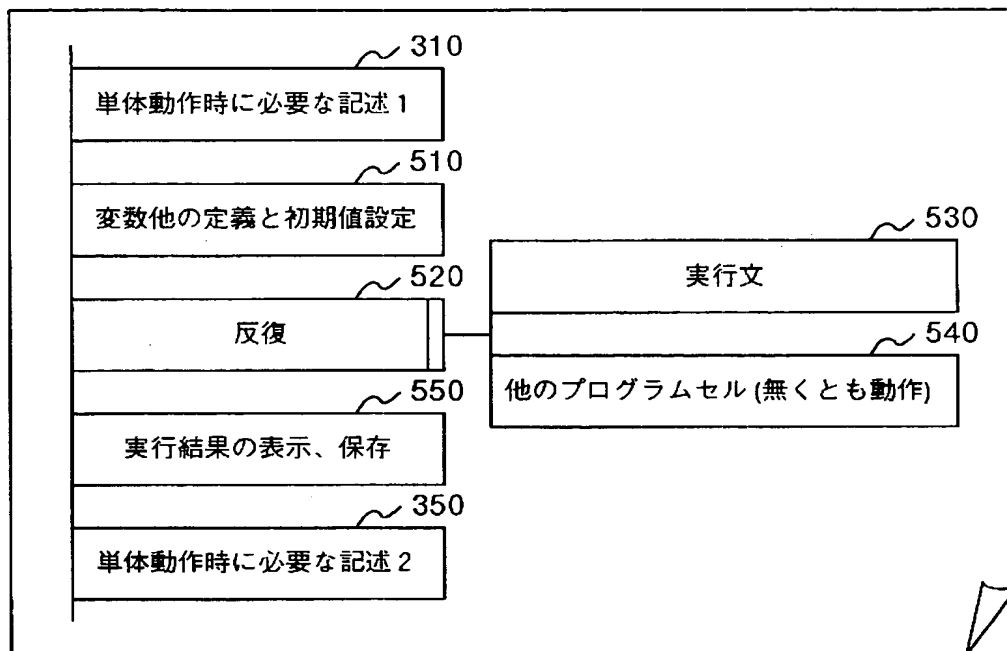
【図 4】

図 4



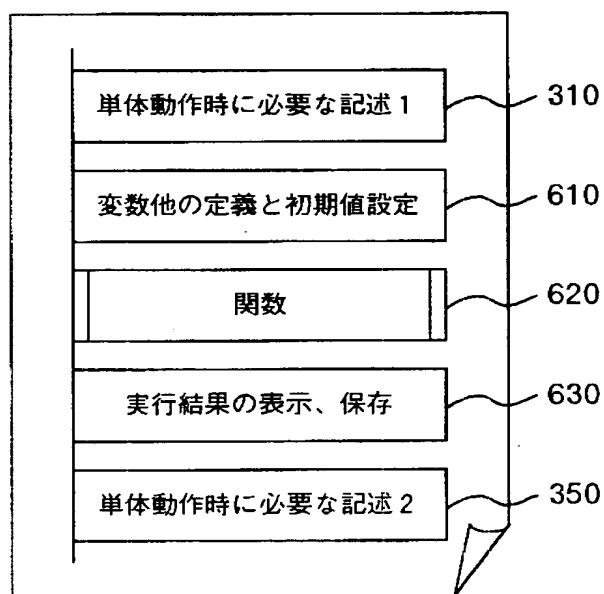
【図 5】

図 5



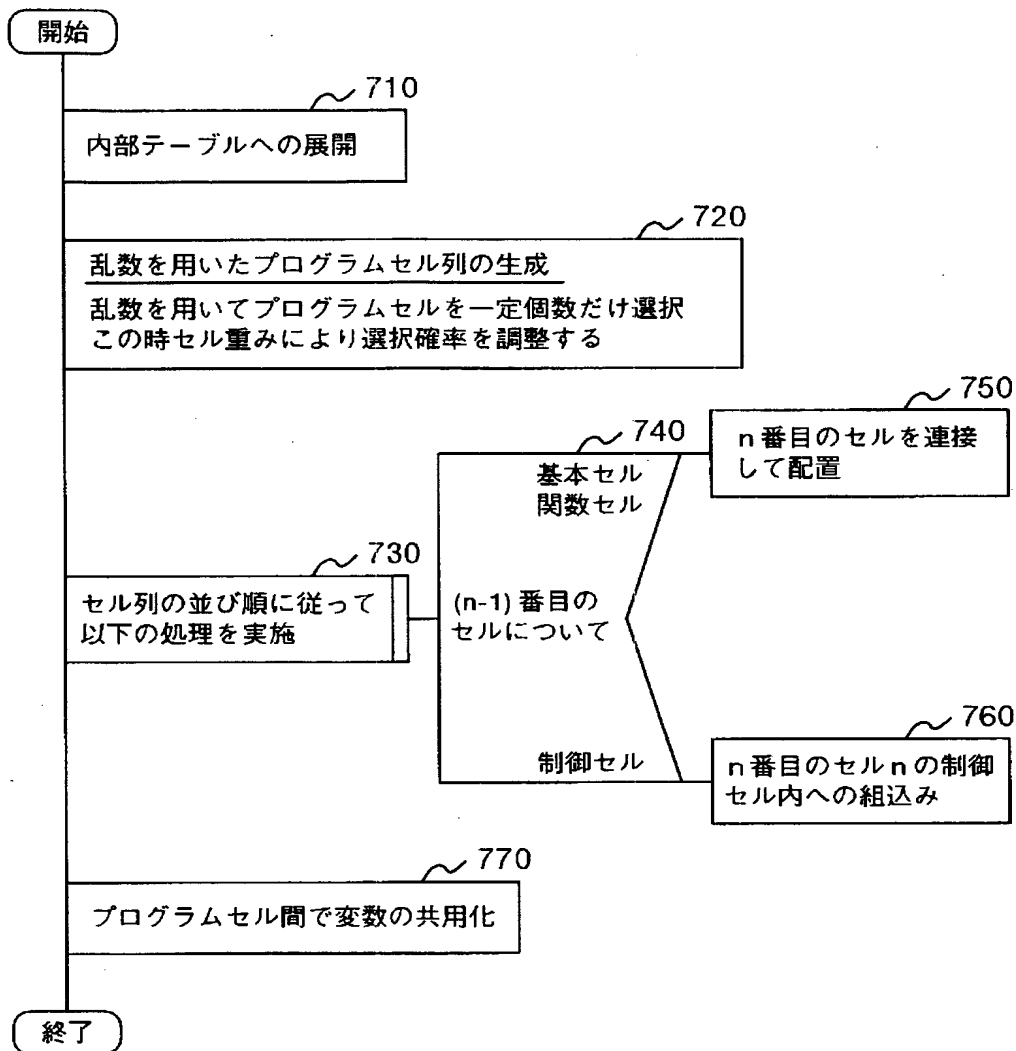
【図 6】

図 6



【図 7】

図 7



【図 8】

図 8

800					
項番	セル名称	セルタイプ	セル重み	セル記述テーブル情報	
				ポインタ	記述長
1	CO1	基本セル	1	0	156
2	CO2	基本セル	3	156	214
3	CO3	制御セル	1	370	81
4	CO4	制御セル	1	451	149
5	CO5	関数セル	2	600	211
	⋮	⋮	⋮	⋮	⋮

【図 9】

図 9

900								
番地	1	2	3	4	5	6	7	...
文字	i	n	t					...

【図 10】

図 10

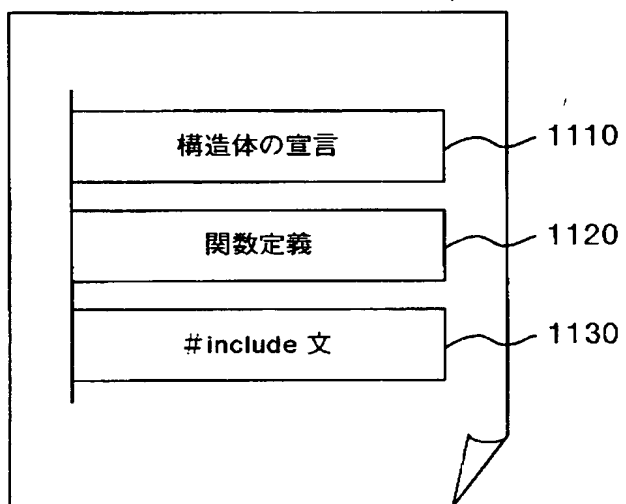
1000

1010 1020

実行順番	選択セルポインタ
1	5
2	3
3	2
4	10
	⋮

【図 11】

図 11



【書類名】 要約書

【要約】

【課題】 計算機アーキテクチャの複雑化に伴い機能的に複雑化したコンパイラをテストするのに、テストプログラムを従来の人手で作成したのでは、その工数が莫大となり、またテスト漏れの惧れがある。

【解決手段】 プログラムセル情報110とセル重み情報111を入力として、乱数発生器120により発生させた乱数を用いて、プログラムセル組合せ器130でプログラムセルを組合せ、コンパイラのテストプログラム140を自動生成する。

【選択図】 図 1

特願 2 0 0 3 - 0 1 1 9 7 0

出 願 人 履 歴 情 報

識別番号

[0 0 0 0 0 5 1 0 8]

1. 変更年月日

1 9 9 0 年 8 月 3 1 日

[変更理由]

新規登録

住 所

東京都千代田区神田駿河台 4 丁目 6 番地

氏 名

株式会社日立製作所

特願 2 0 0 3 - 0 1 1 9 7 0

出 願 人 履 歴 情 報

識別番号

[0 0 0 1 5 3 4 5 4]

1. 変更年月日

1 9 9 6 年 9 月 6 日

[変更理由]

名称変更

住 所

神奈川県秦野市堀山下 1 番地

氏 名

株式会社日立インフォメーションテクノロジー

2. 変更年月日

1 9 9 9 年 8 月 2 5 日

[変更理由]

住所変更

住 所

神奈川県足柄上郡中井町境 4 5 6 番地

氏 名

株式会社日立インフォメーションテクノロジー